DETERMINATION OF LEARNING ARCHITECTURE TO DETECT INT-PHISH PHISHING DETECTION

Submitted By

Fahim Muntasir
(171-35-1900)

A thesis submitted in partial fulfillment of the requirement for the degree of Bachelor of Science in Software Engineering

Department of Software Engineering
DAFFODIL INTERNATIONAL UNIVERSITY
Summer – 2020

Approval

This thesis titled on “**DETERMINATION OF LEARNING ARCHITECTURE TO DETECT INT-PHISH PHISHING DETECTION**”, submitted by **Md. Fahim Muntasir**, **171-35-1900** to the Department of Software Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Software Engineering and approval as to its style and contents.

Dr. Imran Mahmud **Chairman**
Associate Professor and Head
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

--- **Internal Examiner 1**

Name of Internal Examiner
Designation
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

--- **Internal Examiner 2**

Name of Internal Examiner 2
Designation
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Name of External Examiner **External Examiner**
Designation
Name of the Department
Name of the University

DECLARATION

It hereby declere that this thesis has been done by me under the supervission of Nusrat Jahan, Senior Lecturer, Department of Software Engineering, Daffodil International University. It also declere that nithor this thesis nor any part of this has been submitted elesewhere for award of any degree.

Muntasir

Md. Fahim Muntasir
Student ID: 171-35-1900
Batch: 22
Department of Software Engineering
Faculty of Science & Information Technology
Daffodil International University

Certified by:

Nusrat Jahan
Senior Lecturer
Department of Software Engineering
Faculty of Science & Information Technology
Daffodil International University

ACKNOWLEDGEMENT

First of all, all admire and thanks to the Almighty for whom my thesis have been done without any superior interruption. I would like to impulse my deepest thanks to my Advisor Nusrat Jahan madam for her kind support and counsel in my work. She helped me to solve whatever any problem. Finally I wish to thank my parents without their love, all through support it may not be conceivable. With their constant encouragement, support and supplication presently I am on the verge of my undergraduate degree.

Table of Contents

List of Figure

List of Table

ABSTRACT

Phishing is a deceptive culture and a shape of cyber-attack schematic which evolved with the sole intention of collecting confidential information by containing the camouflage of the original website. Most of the people lead a broad range of business via online, they can offer and purchase merchandise, perform diverse banking deeds and indeed take part in political and social selection through online vote casting. Neither purchaser nor vendor needs to meet for any type of transaction and a purchaser can in some cases be trading with a deceptive business that does not really exist. An ordinary hazard comes from reputed phishing websites, which have become an issue for online banking and e-commerce clients. Phishing websites endeavor to trap individuals into uncovering secure data in order for the fraudster to get to their accounts. **The websites that look like legitimate entities used for users who lack knowledge of browser clues and security indicators.**

The aim of the study is to propose an intelligent framework to detect phishing URLs which generates a scientific report by evaluating various multi-layer approaches. This scientific report provides information on the best architecture for phishing URLs detection and also helps anti-phishing tools developers to make an initial decision about approach that should be followed.

This paper proposed a novel phishing URLs detection architecture using a) Deep Neural Network (DNN) b) Neural Network (NN) c) Stacking. In the first level, stacking base classifier provides temporary prediction along with cross validation and crisps prediction. After the completion of the cross validation, the second level requires another additional classifier called meta-estimator that is used in the train set and performed on a test set for final prediction. Neural networks work well with this dataset for better training, time and complexity. Two types of neural networks are used for neural network architecture, five layers are used for deep neural networks and two layers are used for artificial neural networks. Optimized parameters have been used for neural network architecture, along with five types of adaptive learning optimization algorithms, in combination with which a better result is selected.

In the case of five-layer Deep Neural networks along with 50 epochs can provide higher accuracy of 0.95, the minimum mean squared error of 0.30, and also a minimum error rate of 0.074. Using two-layer neural networks along with 150 epochs can provide higher accuracy of 0.95, the minimum mean squared error of 0.29 and also a minimum error rate of 0.07. Stack generalization can reach maximum accuracy 0.97 in binary classification and also provide minimum error rate MAE 2.1.

Machine learning approaches were utilized to identify the modern as well as the variation of malicious URL viably. In any case, by the advancement of exploration in machine learning-based inquiry about, it can be observed that deep learning-based architectures performed better in comparison to the machine learning algorithm.

Keyword: Uniform resource Locator (URL), Phishing, Deep Neural Network (DNN), Neural Network (NN), Stacking

CHAPTER 1

INTRODUCTION

1.1 Background

Phishing could be a felonious component that utilizes both social engineering and specialized subterfuge to take consumers' individual identity data and budgetary related account information and budgetary account credentials of clients **(**Huang et al., 2009). (Cui et al., 2017) proposed to detect phishing websites through a hierarchical clustering approach which bunches the vectors produced from DOMs together concurring to their corresponding distance. A few considers centered on detecting phishing URLs by using the potential characteristics of URLs. One to two hidden layers are usually used for neural networks. In some cases of deep learning, the number of layers varies. But it requires nearly more than 150 layers (Le et al., 2018). There are a few rules to decide the number of layers that incorporate two or less layers for basic data sets and for computer vision, time series, or with intricate datasets extra layers can give way better results **(**S. S. M. M et al., 2021**)**. Phishing is an act like criminal offence, in which the attacker attempts to retrieve confidential and sensitive information about the bank account or social media account in order to trap the end client. Clients when visiting the outlined URL at that point give their secret data and this function can harm the client in different distinctive ways **(**Parthasarathy et al., 2016**)**. Mostly classification the data patterns are accessible in a structured way. But the URL information isn't accessible in a settled pattern. Applying the classification methods or machine learning techniques in URL data. In this way additional approaches ought to be utilized for overseeing the URLs **(**Woogue et al., 2017) Phishing could be a pivotal issue in web security. Phishing detection technique Enables URLs recognition through Various URLs evaluations. Apropos assess the URLs, a number of procedures are accessible. Among the accessible techniques the machine learning techniques are more compelling and precise. Such techniques the malicious URLs patterns become acquainted by classification algorithm and when requisite. It distinguishes the URLs sorts that are phishing or legitimate **(**Dong et al., 2015)**.**Phish tank database is a norm assortment that keeps track of phishing reported URLs by various web security organizations. This database stores a variety of features **(**Mohammad, 2016)**.**

1.2 Motivation of Research

The amount of phishing attacks has been creating amazingly as of late and is considered as one of the foremost dangerous present day web infringement, which leads people to lose belief in web business. Hence, it has a colossal negative effect on online exchange, advancing endeavors, associations' profit, associations, consumer, and by and huge commerce deeds. In this thesis we are going to compare different multilayer approaches to generate a scientific report that provides an optimized architecture and develop anti-phishing tools. An anti-phishing tools developer can make an initial decision from here.

The detrimental impacts of phishing may be to a degree to get to the clients' secret subtle elements, which may result in budgetary misfortunes for clients and indeed avoid them from ingress their personal accounts. Consequently in this study, an optimized architecture will be provided to anticipate and moderate the hazard of phishing websites.

1.3 Problem Statement

Usually, the components of a phishing site are literally and outwardly comparable to a few legitimate sites. The security challenges facing today due to phishing are growing rapidly. Conforming to an eminent Washington based cyber Security Company F5 Systems, Inc. expressed in a report that, the technique of a phishers incorporates three particular missions stamped as Target choice, sociology and Technical manipulation (Pompon et al., 2018). According to the Anti-Phishing Working organization, there were 18,480 momentous phishing attacks and 9666 curiously phishing regions in March2006. It impacts billions of site clients and enormous costing boundaries to businesses (Viktorov, 2017). The prospective expenditure of computerized offense to the around the global network could be a phenomenal 500 billion USD and a clue break will fetch the ordinary organization around 3.8 million USD expenditure, considering that evidence by Microsoft, in 2018.

There is a lot of technique to be done to detect phishing attacks. In this paper use the DNN, NN, and Stacking technique. The aim of this study is to propose an optimized architecture. Basically, the following problems are going too implemented in this study:

- Find out the best Machine learning classification algorithms for the features
- Find out the best optimized architecture to be used for detecting phishing attacks.
- Detecting the best combination of adaptive base optimizer algorithm for neural network and deep neural network
- Multilayer technique stacking will be used here for better performance of machine learning algorithms.

1.4 Research Question

- How rule-based optimized architecture is proven to be more accurate in predicting the phishing website

1.5 Research Objective

The objective of this expedition to instruct a relative evaluation among different multilayer approaches, proposed to an optimized architecture and from here a scientific report is generated. The objective of this study are given below:

- Phishing URLs detection has been implemented to improve the accuracy by the stacking concept.

- Combining all types of classification can perform phish stack, like machine learning, ensemble learning and neural network based approach as base classification.
- Expressing intelligent Anti-phishing architectures with a thorough approach to expand anti-phishing approaches.
- Effect of learning rate in neural network-based technique.
- Appraise of training accuracy with regard to mutate in learning rate
- Detecting the optimized parameter that are suitable to develop the result for neural network
- Detecting the combination of adaptive learning optimization algorithm with neural network
- Detecting the comparison among multilayer approach
- Generate a scientific report to detect phishing URLs

1.6 Research Scope

This work is mainly done for the anti-phishing tools developer. The scope of this research is to propose an optimized architecture for phishing URLs detection. If anyone goes to anti-phishing tool development from here will get a strong basement for tools development. Here a Scientific report generated based on the comparison of some multilayer approaches which will help an intelligence anti-phishing tools developer to make an initial decision. Recently several researchers have proposed different systems for detecting phishing URLs because of the need for legitimate algorithm area the execution of that framework can be influenced. This study uses three types of multiple approaches: DNN, Neural Network (NN), Stacking technique. Some optimized parameters are combined with these multilayer approaches for better performance (Vrbančič et al., 2019). A comparison among these multilayer approaches is proposed to an optimized architecture and from here a scientific report is generated. By studying this scientific report, an intelligence anti-phishing tools developer will gain a thorough knowledge of all aspects of phishing tools.

1.7 Thesis Organization

Chapter 1: Introduction: Background, Motivation of Research, Problem statement, Research Question, Research Objective, Research Scope, and finally thesis organization. Chapter 2: Literature Review: Phishing, Stacking, Neural Network, Deep Neural Network. Chapter 3: Research Methodology: Data Collection, Comprehend URLs, Feature description, Data pre-processing, Model Generation phase, Deep learning algorithms, Formal Information about stacking, Adaptive Optimizer, Machine Learning Algorithm . Chapter 4: Result and Discussion: Evaluation Parameters, Experiment Result. Chapter 5: Finding and Contribution, Recommendation for future work.

Chapter 2

Literature Review

2.1 Phishing

(Adebowale et al., 2019) proposed an ordinary technique that there are some users who steal confidential information from websites and call those users are phishing users. This activity commonly happens by fake websites or malicious URLs that's called fraudulent ventures. Cybercriminals use fraudulent activities to create a well-designed phishing attack. Gaining access to the victims systems the cybercriminals could install malware or inapt protected user systems.

(Vazhayil et al., 2018) anti-phishing working groups anti-phishing working group have released the reports that the number of phishing websites increases every month focusing more than 450 brands. Exponential development in the number of phishing websites, blacklists has its own limitations. Machine learning or deep learning techniques can be used to detect newly generated phishing URLs.

(Adebowale et al., 2019) claimed that the critical threat of web activities is phishing. At that point for gathering the personal data of online victims the attacker mimics the original website of an organizational formation.

(Acquisti et al., 2017) suggested that reduce the threat of phishing assaults, Indicating at directing the hazard of phishing attacks, various strategies are recommended to get ready and instruct end users to recognize phishing URLs

(Maennel et al., 2017) claimed that the accomplishment of phishing website recognizable proof methods basically depends upon seeing phishing destinations absolutely too, interior a palatable timescale. Various conventional strategies subordinate on settled exceedingly differentiating posting databases have been prescribed to recognize phishing destinations.

2.2 Stack

(Li et al., 2019) removed features from URLs and hypertext The markup language (HTML) of the suspicious site. The Elevated features include 8 URLs and 12 HTML-based features to create feature vectors. It was fuelled by a stack model equipped for classification and accomplished an accuracy of 97.30% and offers a combination of a stacking show combining Gradient Boosting Decision Tree, LightGBM, XGBoost Algorithm for detecting the phishing web pages.

(Z. Q, D, 2017) suggested ensemble classifiers for e-mail filtering that excluded five algorithms that’s Support Vector Machines, K-Nearest Neighbor , Gaussian Naive Bayes, Bernoulli Naive Bayes, Random Forest Classifier. Ultimately random forest was improved accuracy 94.09% to 98.02%.

(Gupta, S. and Singhal, A., 2018) proposed that approximately for minimum execution time random forest tree is an admirable strategy to detect phishing urls.

(Parekh et al., 2018) proposed a different technique for phishing sites detection using random forest and claimed that the accuracy is 95% proximately as maximal.

2.3 Neural Network

(Vrbančič et al., 2018) recommended setting parameters of deep learning neural networks that are swarm intelligence based techniques. After that the proposed technique applied to the classification of phishing website and capable of better the detection by comparing to existing algorithm

(Vazhayil et al.,2018) proposed that developing architecture used Neural Network (NN), Support Vector Machine (SVM),Biased support vector machine (BSVM) and Self Organizing Map (SOMs). Set of features are number of sub-domains, domain age, HTML formatted emails, number of domains, IP based URL.

(El-Alfy, E. S. M., 2017) recommended a framework that connected unsupervised and supervised algorithms for training the nodes. Phishing sites depend on feasibility neural networks and clustering K medoids. Feature selection and module is used to reduce space capacity is used by K-medoid technique. Thirty features are achieved 96.79% accuracy by the desired technology.

(Mohammad et al.,2020) proposed that ANN techniques have been selected for appraise Precision and RMSE conditions.The DOM tree should be played out to upgrade the trustworthiness of the component vectors, frst, an examination of the topology structure of the site as display. Ball support vector machine (BVM) classifier analyzed to recognize the element vectors.

(El-Alfy, E. S. M., 2017) analyzed the productivity of malicious web page detection using artificial neural network (ANN) technique with static classifiers such as SVM, DT, NB and KNN by using the static feature sets from lexical in URL and page contents. Observation to other static classifiers ANN provide better execution by maximal accuracy of 95.08%. Discussing the particular importance of each feature towards recognizing attacks and in that way decrease the false positive rate.

2.4 Deep Neural Network

(Le et al., 2018), recommended to DNN, are trained with implied deep stacking. The evaluated covers of the past outlines are upgraded as it were at the conclusion of each DNN preparing epoch, and after that the upgraded evaluated veils give extra inputs to train the DNN within the other epoch. At the test period, the DNN makes expectations successively in a repetitive manner. In expansion, we propose to utilize the L1 loss for training. Implicit.

(Yang et al., 2019) claimed that formed on aforesaid, investigate to the detection of sociology attacks associated with phishing URLs, using Deep Learning techniques. Union of each preferred work and the classification of anti-phishing formula through its way, getting that the URLs-oriented way is the foremost utilized. The aim of the study is arranged to deliver a total vision to the strategies of relief of phishing URLs by implies of deep Learning algorithm. Moreover, to classify the Deep Learning algorithms chosen in each explanation, That succumb that the foremost commonly utilized are the DNN and convolutional neural arrange (CNN), among other principal information.

(Kumar & Indrani, 2020).propose that Fuzzy Deep Neural Network classifiers achieve the most excessive accuracy in the prediction analysis of phishing, legitimate, and suspicious URLs. Using Association Rule mining based on these features then seventy-five optimal rules are produced. Using DBA-based detector segments the optimal features are extorted from the datasets. Beat this issue, the frequent rule reduction algorithm along with the classification technique for forecast Phishing websites should be applianceed.

(Winterrose et al., 2020) claimed that exploring distinctive properties of veritable oversees methodologies for recognizing phishing web goals. Phishing URLs utilizing significant learning strategies, for case, profound Boltzmann machine (DBM), stacked auto-encoder (SAE), and profound neural organize (DNN). DBM and SAE are utilized for pre-preparing the show with a predominant depiction of information for attribute assurance. DNN is utilized for twofold gathering in recognizing darken URL as either a phishing URL or a genuine URL. The proposed system fulfills a higher area rate of 94% with an undermost false positive rate than other machine learning procedures.

(Vrbanˇciˇet al., 2019) tending to the issue of utilizing swarm intelligence algorithms to parameter setting for a deep neural network. Connected the proposed strategy variations to the classification errand for recognizing between phishing and legitimate sites. The execution of the proposed strategy is assessed and compared against four distinctive phishing datasets. Compared to the manually tuned deep neural network, we were able to statistically significantly improve the predictive performance by utilizing the proposed swarm intelligence based methods by utilizing the proposed swarm intelligence based methods. Above all, whereas the advancement of F1-score come to indeed 24% on one of the datasets the enhancement of classification exactness ranges from 2.5% to 3.8%,

(Sahingozet et al., 2018) pointed to utilizing machine learning based algorithms, Artificial Neural Networks (ANNs) and Deep Neural Networks (DNNs), to prepare the framework. Analyzing the URL of web pages to catch abnormal requests. Using a dataset which contains 37,175 phishing and 36,400 legitimate to train the model.as reported by the experimental outcome, the use of ANN and DNN approaches approximately the accuracy in detection of phishing websites with the rate of 92 % and 96 %

CHAPTER 3

RESEARCH METHODOLOGY

3.1 Data Collection

A publicly accessible dataset has been used for training or creating the architecture. The initial part of this model is to collect data and analyze the datasets. This dataset was collected from the UCI repository. It has a total of 11055 different types of URLs. It has a total 30 features used to train the model (S. S. M. M et al., 2020). The following table describes various aspects of this dataset.

Total features of dataset	**30 features**
Total URLs	11,055 URLs
Phishing URLs	4898 URLs
Legitimate URLs	6157 URLs

Table 3.1: Dataset Information

How the information was collected to get this dataset, we ought to get the URLs that are marked out underneath.

3.2 Comprehend URLs

Need to have the whole idea of the different parts of URLs. In the event that you notice the attack patterns, able to see that URLs play an imperative part. For the most part, a URL comprises of five essential element which is portrayed in the figure below:

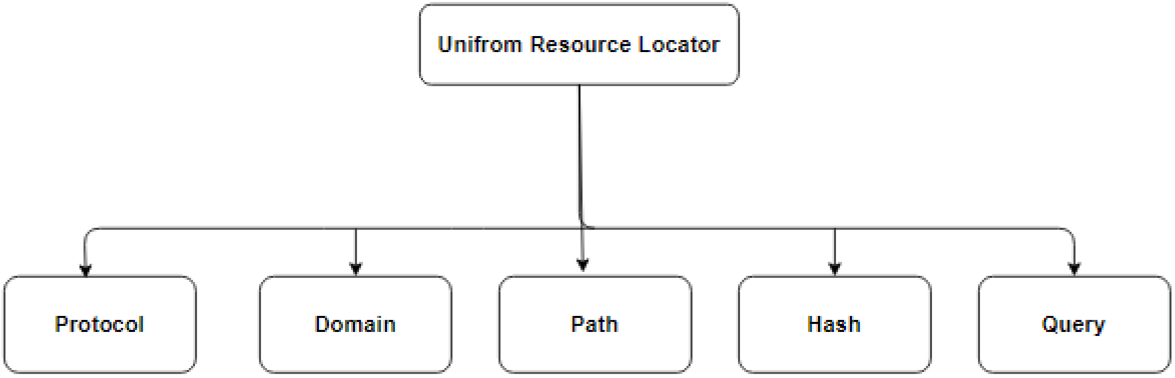

Figure 3. 1: Five basic components of URLs

The beginning portion of the URL is called Protocol that may be a set of acts utilized for exchanging information. Second portion of the URL is called domain which has a few parts. To begin with, a portion of space regularly contains the "www" prefix that alluded to as World Wide Web Address. At that point, the title of the site is given after a dot. After a dot, we include the corporation sort taken after by the nation code in case included. Third part of the URL contains the way points of interest which addresses the segment and page of the site. Fourth portion comprises Query which alludes to a portion of a page. At last the Query portion sends extra data sent with the page request.

3.3 Feature Description

Ought to analyze features and the possibility of working with these features, sometime recently venturing into the feature selection portion. Essentially, in this dataset there are almost four fundamental features which have in add up to 30 sub-features. Formed on the information, each single feature gives data almost whether the site can be phishing, legitimate or suspicious. In this segment, we are planning to point up the features.

3.3.1 Address bar-based features

The address bar that means URL bar or location bar could be a GUI gadget that appears in an ongoing URL. According to the dataset it has 12 sub-features. That is appeared on the table 3.2 underneath

3.3.2 Abnormal Based Features

It for the most part centers on abnormal exercises on the site. According to the dataset it has 5 sub-features. That appears on the table 3.3 underneath.

3.3.3 HTML and JavaScript based features

According to the dataset it has 5 sub-features. That appears on table 3.4 underneath.

3.3.4 Domain based features

Using domain names prepares effortlessly identifiable and unforgettable names numerically. According to the dataset it has 7 sub-features. That appears on table 3.5 underneath.

Name of the features	Explanation
Ip Address	In the event that IP address is utilized as an elective of a domain name within the URL that is a phishing website and client can almost be sure somebody is attempting to take his credential data . From this dataset, discover 570 URLs having an IP address which add up to 22.8% of the dataset and proposed a rule IP address is in URL that called Phishing, otherwise its Legitimate
Length of URLs	Long URLs are mostly utilized to cover up the dubious portion within the address bar because it contains malicious content. Deductively, no well-founded length that recognizes phishing URLs from legitimate ones. For that legitimate URLs proposed length of the URLs is 75. In this study to guarantee the accuracy measured the length of URLS is suspicious, legitimate or a phishing site in this dataset and proposes an average length. From this proposed condition the URL length is less than or equal 54 and it is classified as legitimate, if the URL is larger than 74 then it is phishing. According to the dataset found 1220 URLs that's length greater than or equal 54
TinyURLs	For shortening the URL length tinyURL is used. It diverts to the most page to click the shorter URL. This interface is like a phishing site since rather than an authentic site it diverts the end client to fake sites.
Operate the @ Symbol	Web browsers mostly ignore the segment that is attached with @ symbol. Because it is kept away from real addresses. According to the dataset, finding 90 URLs that have the '@' symbol will add up to only 3.6%.
Operate the "//" symbol	After HTTP or HTTPS the "//" symbol is used as legitimate URLs. On the off chance that after the initial protocol statement that's considered phishing URLs. //" symbol is utilized for diverting to other sites.

Domain names prefix or suffix separated by “-” symbol	If any URL contains the “-” symbol in its domain name then consider it's a phishing URLs. Generally validated URLs don't contain the “-” symbol.
Operate the “.” symbol in domain	When a sub-domain with the domain name is added, it has to include dot. Considering suspicious in case drop out more than one subdomain and larger than that will point it like a phishing
HTTPS with secure socket layer	Most of the legitimate site HTTPS protocol and the age of certificate is exceptionally vital for using HTTPS. For this that's need a trusted certificate.
Expiry date of domain	Principally domain name have longer expiry date for legitimate sites
Favicon	Favicon can divert clients to suspicious sites, when it is stacked from outside space. It's by and large utilized in websites and it's a graphic image
Utilizing insignificant ports	Phishers continuously discover defenselessness and attempt to require an advantage on the off chance that any URLs has some open ports that's superfluous.
HyperText Transfer Protocol in domain	The phishing websites are considered if any URLs of this website have HTTPS on domain name

Table 3.2: Address bar-based features

Name of the features	Explanation
Request URL	From another domain on the off chance that a page contains larger amount of outside URLS that's considered it suspicious or phishing
Having URL of anchor	Comparable to the request URL features, the chance of phishing increases, more <a> tags utilized inside the site.
Link among (Meta, script, Link) tag	It is calculated as either suspicious or phishing formed on their proportion if the tag contains large number of outer links
Server form handler	Phishing is considered in case the Server shape handler is blank or empty. Server frame handler diverts to a distinctive domain It's checked as suspicious.

Having an email to submitting information	It is considered as phishing, rather than a server, web form coordinated to an individual email is submitted the information.
Abnormal URLs	It considered as phishing, In case the character isn't included within the URLs

Table 3.3: Abnormal based features

Name of the features	Explanation
Forwarding website	It can be frightening, on the off chance that diverting is happened different times
Customization of status bar	To alter the status bar of the URLs can be utilized on ``Mouseover" occasion. It continuously appears off genuine URLs and stows away the fake URLs. at a time When it's connected on the site that's obliging as phishing
Right click disabled	Users can't check the source code; right-click functions are impaired mainly by Phishers. When the framework is debilitated within the site that's obliging as phishing.
Having Pop-up Window	Pop-up window with a text field is consisted by a web page that's obliging as phishing
Custom IFrame	Stowing them away within the website phisher could be utilized IFrame. In for the most part Connect outside substance to appear in a domain utilized by IFrame.

Table 3.4: HTML and JavaScript based features

Name of the features	Explanation
Age of Domain	Obliging a authentic site as phishing site tend to live for shorter period of time in the event that the age of domain is longer than six month
Record of DNS	It is exceedingly recommended as phishing site within the event DNS record isn't contained by website
Traffic of website	Colossal amount of individuals visit websites for the most part because it would have higher positioning. Positioning can distinguish on the off chance that a location is phishing or not. A phishing site is being tends to have a lower chance by the next ranked site
Ranking of page	In most time that phishing websites have no PageRank value since this value is allotted on its importance
Indexing of Google	A legitimate site can be accepted by a site that has a title on the google index.
Reports Statistical	Guessing it as phishing webpage within the event the have of the webpage has a place in any beat phishing IP's or domains
Joins indicating to Page	Phishing site prohibiting have much links indicating apropos it since it has shorter lifetime

Table 3.5: Domain based features

3.4 Data Pre-processing

The foremost imperative component of a thesis is the dataset. Inside real life most of the dataset are deficient and the dataset has missing values. For the future execution within the recommended model it's barely ought to standardize data.

When training the dataset, split the dataset into two parts: the training portion and testing portion. Training portion of the data gives more precise results. Test portion of the dataset which might diminish the accuracy of the outgrowth. Splitting data here uses a k-fold procedure. The reason for utilizing the k-fold procedure is portrayed underneath.

- Reducing conflict, less overfitting and models like general for that reason shuffling the data randomly
- Splitting the entire dataset into a group.
- With the aim of each group put the group being test dataset, training dataset, create a system formed on training dataset, evaluate the result based on test dataset, hold the assessment score and dispose of the system.

- Eventually epitomize craft the system and requite the score.

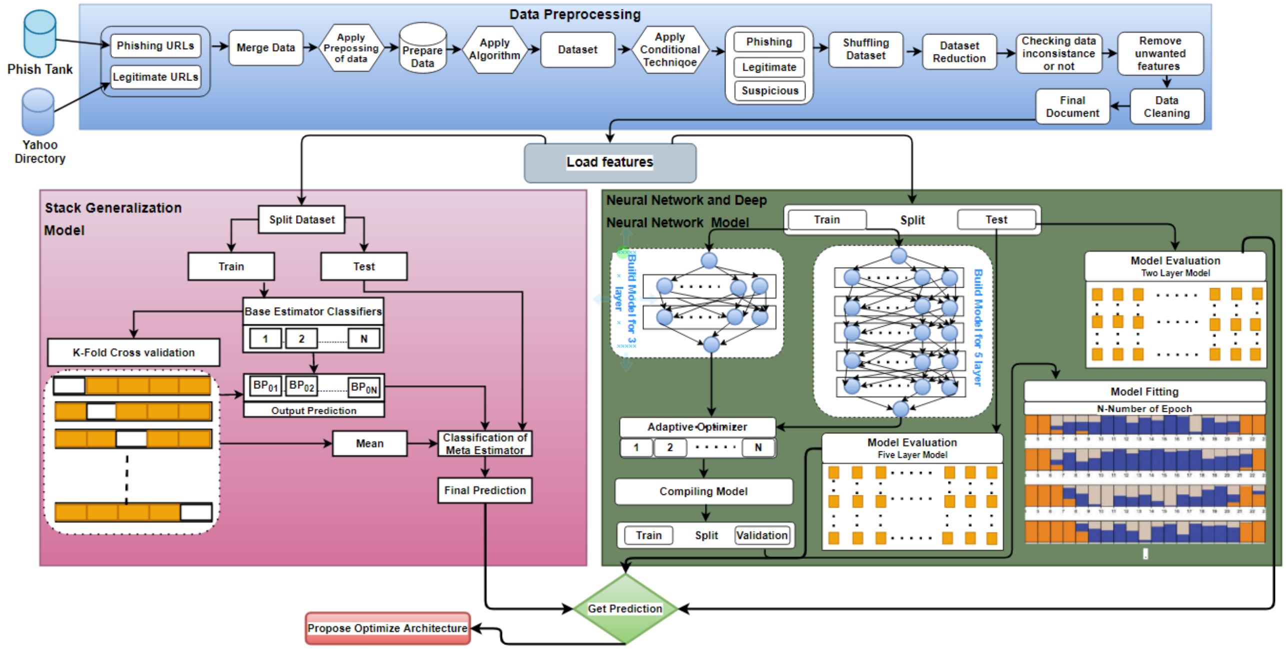

Figure 3. 2: Methodology

3.5 Model Generation Phase

The above methodology consists of three parts. The first part shows the method of data processing. The next parts are NN, DNN and stack generalization. The main purpose of this model is to determine the best output through evaluation by applying stacking technique and neural network and deep neural network on the processed data set and to propose an optimized model based on that output. First of all some phishing URLs are collected from phishtank and some legitimate URLs are collected from the yahoo directory to create the data set. The phishing URL and the legitimate URL are then merged together. In this way apply preprocessing techniques according to the data processing model and from there data is prepared. A number of algorithms is applied to this generated data then a data set is obtained. Some conditional techniques are applied on this data set based on which the values of the data set are divided into three parts: phishing, legitimate and Suspicious. The next step is shuffling the dataset after this technique applying reduction on the entire dataset and checking if the dataset is inconsistent or not after checking to remove unwanted features. By cleaning the data set get the document of a complete data set. Now this data set is ready to use. Now an optimized output will be provided by applying neural network and deep neural network technique on this data set. After loading the features from the data set, the data set is split into two parts, test and train The train segment is applied to a two-layer neural network architecture and (Somesha et al., 2020) a five-layer deep neural network architecture, respectively. Since the data set is of binary type, for binary classification problem non-linear activation function ReLU is used for hidden layers of neurons and sigmoid function is used for output layers of neurons (Vrbančič et al., 2018). According to this architecture, five types of adaptive optimizers have been used here. The next step is to compile the model using these optimizers. It is then divided into two parts, train and validation, by splitting the train set.. The model is fitted using a number of epochs and early stopping techniques, to prevent overfitting. Now two outputs are available by evaluating the two models using the test set. After applying the approach now apply stack generalization technique in the dataset. The evaluation technique of stack generalization has been described in figure. It's a multilevel approach. Stacking is usually done in two steps. In the first level stacking provides transitory prediction using base classifiers with k-fold cross validation and output probability prediction are revealed. During the system formation the output prediction and transitory prediction of step one are used in second steps. The estimate theory of phish stack are described below (S. S. M. M et al., 2020):

- In the first step of stacking by using base classifiers to predict train and test set according to the second step the desired predictions are being acquired then that are considered as features.
- Stacking is a multilevel approach so any kind of algorithm can be used to predict it in two steps
- This proposed system used k-fold cross-validation so that it eluded overfitting for this training set and each fold of the train portion it may predict using out-of-fold. According to this proposed model the value three to ten is used for k-fold cross validation after all provides output using a test set.

In the first step at the end of training the data the output is predicted using the test set. This time it's complete with all folds technique that's needed to mean for estimating all values from all folds that are used.

In the second step connected to another classifier that's called a meta-estimator on the train set, from the test set it performs terminal prediction. This approach takes extra time because it again adds a classifier for its performances. When the k-fold cross validation done in the first step then prediction is not completed these are completed on the second step.

Three outputs are obtained from the above multilayer techniques then a model is selected based on the decision, according to the value of the output. An optimized architecture is proposed based on that model.

3.6 Deep learning algorithms

Against numerous different parameter alliance feature sets has been trained and used cross-validated to assess the execution of the feature set. In phishing site classification to achieve the greatest accuracy must collect information formed on feature sets, tune the hyperparameters (Vrbančič et al., 2018). In this way training networks need to set hyperparameters and authenticate across suitable treasure. Highest probability can be simply classified by phishing sites, afterwards achieving the appropriate treasure. Deep learning algorithms are implemented by applying used python and tensorflow many combinations of hidden layers show that five hidden layers provide the best result for DNN. Features represented most effectively because it extracted in the complex functions, nonlinear, separable. The suggested deep neural network comprises 2 layers, with (Somesha et al., 2020) 5 hidden layers, one input layers and one output layers. Rectified Linear Unit (ReLU) or sigmoid function taken after and standardized all layers. In this five layer to begin with three layers utilizing ReLU function and the output layer taken after sigmoid work. Early stopping is speed up training to reduce the over-fitting and internal covariate shift. Dodging noteworthy delays within the rate of gradient descent joining after an initial set of emphasis (Vrbančič et al., 2018), ReLU enactment has supplanted sigmoidal or tanh incitation capacities in secured up layers due to its inclination to memorize speedier than sigmoidal or tanh.

3.6.1 Formal Information about DNN

DNN is composed of numerous common neural network layers. In order to describe the architecture of neural networks, approximately one input layer, one output layer and at least one hidden layer have been taken. It is shown in the following figure 3.1.

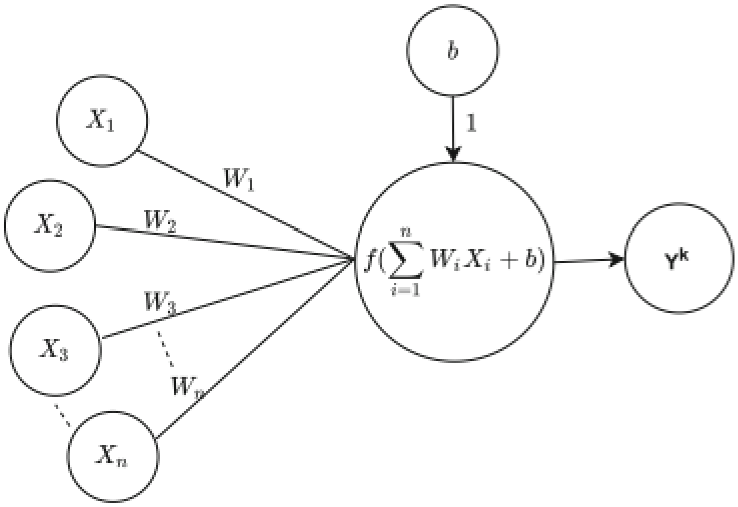

Figure 3. 3: Simple neural network architecture

Biological neurons that perform mathematical techniques for the capacity of information motivate neurons. Data is transferred to other neurons. Each layer is made up of essential computing units, subsequently data aggregators within the neural network. A neuron's numerical represented (Li et al., 2019) is

$$Y^k = \Phi\left(\sum_{k=0}^{k=n} W_{kj} x_j + b_k\right)$$

From this above equation activation function is Φ. $W_k \in R^{LB}$ is weight of K^{th}

Neuron, Y^k is the output of K^{th} neuron. Numbers of neurons in the input layer based on the size of the dataset and the number of features. In this equation $X \in R^{LK}$, here L is the entire number of the datasets, K is the entire number of features in datasets and R expresses the real number. Output layer of neurons depends on how many outputs are required for the proposed system (Vazhayil, et al., 2018). In the hidden layer there number of neurons is called hyperparameter that should be settled to acquire maximal outcome. Sometimes neuron identifies difficulty in view of the fact, neuron performs estimate. DNN holds itself pursuant to the data because it's an intricate statistical technique.DNN executes remarkably with train data on the other hand it's not remarkable with new data. If making the network system critical then the desired outcome could possibly be over-fitting. There are five hidden layers in the deep learning model and Y is the input layer of{ 1,2,3,4} and also Y is the output value of this layer . The weighted layer is W. For linear transformation used I of inputs from n layers to and output of m layers. Activation function of each layer is F, bias of the layer is B so at last two one is Y^0 is used for the input layer and Y^1 is the output layer (Li et al., 2019).

$$Z^{(l)} = Y^{(l-1)} * W^{(l)} + B^{(l)},$$

This equation have taken from (Rao & Ali, 2015)

$$Y^{(l)} = F(Z^{(l)})$$

Multiplication matrix is *Where the value of W splits with Xavier Initialization the primary value of B is zero and backpropagation method, after each iteration the value of W and B are updated after for each iteration. By using ReLU function hidden layers are activated in this above equation I recommended Ith iteration and L recommended Lth layers. By utilizing sigmoid activation technique the intervening outcome of the system is Y (Li et al., 2019). .

$$Y^* = \frac{1}{1 + \exp -Z^l}$$

In the total dataset loss function defined as (LY*; Y$^\wedge$) and output layer is 1. This total dataset specifies the whole cross entropy within one outputs like model and real that is shown as below (Li et al., 2019). .

$$L(Y^*, \hat{Y}) = \sum_{j=1}^{n} [\hat{y_j} \log y_j^* + (1 - \hat{y}_j) \log(1 - y_j^*)]$$

Following the processing through deep learning technique, In this total dataset Y is an intervening outcome. Actual level of the dataset is $Y^*{}_j \in(0,1)$ is jth row of Y^* and $y^\wedge$ is the jth row of while Y^ is an actual label of the entire dataset and $y^\wedge{}_j \in(0,1)$ g is the jth row of $Y^\wedge$, Here legitimate and phishing sites are expressed by 0 and 1. Using Optimizer at every epoch to update parameters optimized by the loss province and this stage deep neural model is trained. Without overfitting these features by province patterns.

3.6.2 Formal Information about NN

NN involves many neurons and have been around for over a decade, for the actual valued activation of each neuron yielding a pattern. The first layer is called the input layer according to the structure of the neuron NN build up neurons, when it activates the sensor everything can be perceived from the environment. Previous layer outcome obtained to be the weighted input to the following layer, there is no correlation among each layer but NN shows craved conduct is made finding the correct weight by knowing NN (Fister et al., 2016) In view of the fact back-propagation trained NN with a master based learning approach to play a major part like gradient descent algorithm when 1980. In the evaluating stage the training method of NN can take a long empowering chain. Total actuation of NN each stage change that's concurrent to non-linear way and apportion credit over numerous such stages (Vrbančič et al., 2018). Feed-forward NN comprehends fixed-size input to a settled measure output; numerous deep learning techniques utilized this approach. Transform from one neuron layer to another has to be weighted esteem; this esteem comes from the inputs of the previous layer weighted sum is measured and through its nonlinear function. The foremost acknowledged non-linear work is ReLU half-wave rectifier f (z) = max(z, 0).tanh(z) or 1/(1 + exp(−z)) generally used as non-linearities. Within many hidden layers ReLU learns much swift and approved training of deep supervised networks without unruled previous training (Adamson & Smith, 2018) Recurrent neural networks are utilized in NN topology by feed-forward NN. The connection between the consequent layer to neutrons within the going before layer, RNN contains observation of these layer connections. NN never subordinate on exterior inputs but subordinate the previous training iteration (Fister et al., 2016) . RNN sequential inputs are exceptionally capable dynamic techniques, their training strategy has exceptionally tricky.

3.7 Formal Information about Stacking

The main features of stack generalization is that it integrates with low grade models using high grade models and also known as ensemble algorithm. The main target of stack generalization is to develop the result of low grade models (Li et al., 2019). New model is trained by other models that are already trained from a dataset. Most commonly stacking uses simple linear function (mean, median, average etc) to assemble the prediction for other models.

3.8 Optimization algorithm

3.8.1 Stochastic Gradient Descent

The characteristic of SGD is according to the dataset in the event that it's finding any information at that point it takes an endeavor to upgrade the weights value and it's upgrading method is more frequent. According to SGD's characteristic loss for each taring set the system parameters are changed. so the whole dataset takes on 2000 rows it would reestablish the system parameters 2000 times. According to the systems parameter are oftentimes renew parameters that different force it has vacillations in misfortune capacities additionally elevated change

3.8.2 Adagard

The main features of Adagard is that it performs with derivate error technique. All the optimizer have a common issue that each cyclic ordered for all parameters the learning rate are same according to adagard can perform to change the learning rate each time for individual parameter it's called second order optimizer technique According to the leaning rate the given parameter is adjusted based on the given parameter on previous gradients. Average of the squares for gradients that's up to time step t and smoothing term ϵ that get off the segment by zero. The execution of an adagard algorithm is more fiendish without square root work. Adagard essentially does lage recharge for less recurrent parameters as well as little steps for recurrent parameters .

3.8.3 Adadelta

The fundamental convention of adadelta is it's extricate decaying learning rate problem also called it could be an extent of Adagard. According to adadelta it gives impediment of collecting previous gradients for settled estimate w, instead of collecting previous squared gradients and the average of all gradients utilized by moving average.

3.8.4 Adam

According to Adam optimization it performs with momentums by two orders. In order to achieve a careful search reduced the velocity according to a suspicion. Average of previous AdaDelta, Adam used to save an explosive decay and it also median of previous gradients holds an explosive decacy. The values of first momentum are mean and second moment that is represented by M(t) and V(t) and the gradients sequentially are uncentered conflict.

3.8.5 RMSprop

The main concern of RMS prop utilizing an additional parameter decay is the common measure and anticipate it's quick development. It must be anticipate the ϑt to urge greater by extra parameter β but it's decay would less the former weighted value.

3.9 Machine learning algorithms

3.9.1 Support vector machine (SVM)

The points of space can be partitioned into segments with clear gaps as representative of the support vector machine training data that's as wide as conceivable. Current examples are marked within the same space it's predicted area formed on the gap it's dropped into. SVM is memory proficient since viable in elevated measurement space.

```
lib ()

x <- bind(a_train,b_train)
```

```
fitting <-svm(b_train , data = p)

summary (fitting)

predicted_output = predict(fitting, a_test)
```

3.9.2 Decision Tree

The fundamental characteristic of a decision tree is that its classes are given data features collectively and it proposes to classify the data can be used by the pattern of law. According to the decision Tree it's easy to get it and envision and deal with quantitative and explicit information properly.

```
lib ()

x <- bind(a_train,b_train)
fitting <- rpart(y_train , data = p, method="cl")

summary (fitting)

predicted_output = predict(fitting, a_test)
```

3.9.3 Logistic Regression

The main objective of logistic regression is logistic function shown using a single path that probabilities portraying the conceivable results.

```
x <- bind(a_train,b_train)
logisticreg <- glm(b_train , data = p ,family='binomial')
summary(logisticreg)

predicted_output = predict(fitting, a_test)
```

3.9.4 Naive Bayes

Naive Bayes classifier performs well in real-life problems and is formed on Bayes theorem**.**

```
lib ()

x < bind(a_train, b_train)

fitting <-naiveBayes(b_train , data = p)

summary(fitting)

predicted_output = predict(fitting, a_test)
```

3.9.5 K-Nearest Neighbors

K-Nearest Neighbour generally holds tarin data because its a form of lazy learning for that reason by the uncomplicated majority support classification is reckoned.

```
lib ()
x <- bind(a_train,b_train)
fitting <-knn(b_train , data = p, k=9)
summary(fitting)
predicted_output = predict(fitting, a_test)
```

3.9.6 Random Forest

The special characteristic of random forest is most of the time it is more perfect than decision tree to decrease the overfitting, it is also called a meta-estimator and used to improve the accuracy that fit decision trees on different sub pattern (Das et al,. 2019)

```
Fundtion Random Forest(S,F)
P ← 0
for Iε1……., C do
S(i)
P ← P ∪ {h,i}
end for
Return P
```

3.9.7 Ensemble method

3.9.7.1 XGBoost

XGBoost is used for more speed and dependable execution and presently connected in machine learning. It is an execution part of a gradient boosted decision tree

3.10 MLP

The main function of the input layer is to collect signal and send it to hidden layers after processing an output layer to provide a result based on input signal. An unpredictable number of hidden layers exist between input and output layers. That is the main strategy of MLP.

CHAPTER 4

RESULT AND DISCUSSION

4.1 Evaluation Parameters

The whole system was mainly focused on evaluation based on data phishing or legitimate that's identified by binary classification. Confusion matrix, Accuracy, Precision-Recall Curve, Classification report, AUC-ROC Curve, Mean Absolute Error (MAE), Mean square Error (MSE) used to evaluate the performance of this system. The evaluation parameters for assessment are described in the following table below (S. S. M. M et al., 2020).

Assessment Parameter	Assessment Parameters Formula	Statement of the assessment parameter
Mean Absolute Error (MAE)	$\mathrm{MAE} = \frac{\sum_{i=1}^{n} \lvert y_i - x_i \rvert}{n}$	It is the average value of all absolute errors[26]
Mean Square Error(MSE)	$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$	It is the average value of all squares errors

AUC-ROC Curve	For Positive Recall
	TRP = TP/(TP + FN)
	For Negative Recall
	FPR = 1- Specificity =1 - TN/(TN+FP) = FP/TN+FP
		AUC - ROC curve is intrigued with True Positive Rate that belongs on y-axis, in opposition to the False Positive Rate that belongs on x-axis[25]

Precision - Recall Curve	For Positive Precision
	P= TP/(TP + FP)
	For Negative Precision
	N= TN / (TN+FN)
	For Positive Recall
	PR= TP/(TP + FN)
	For Negative Recall
	NR = TN/(TN+FP)
		According to the precision-recall curve for a single classifier, estimating and intrigued the precision in opposition to the recall [24].

Accuracy	Accuracy = (TP + TN) / (TP + TN + FP + FN)	Accuracy means the rate of prediction that model executes [28].
Misclassification Rate	Error Rate = 1 - Accuracy	The failings of identify value that is not appropriate for classification

Table 4.1: Evaluation Parameters

Confusion matrices play an important role in getting results from various classifiers. The confusion matrix has four attributes as described below (Das et al., 2019).

Matrix	Statement
True negatives (TN)	According to TN, it understood that the detected website is legitimate that means phishing as phishing.
False Negative (FN)	According to FN, it understood that the detected website is legitimate, which means phishing as legitimate.
False positives (FP)	According to FP, it understood that the detected website is Phishing that means legitimate as phishing
True positives (TP)	According to TP, it understood that the detected website is Phishing that means legitimate as legitimate

Table 4.2: Four attributes of Confusion Matrix

4.1.1 Precision

Precision holds the positive predictions accuracy.
Precision = TP / (TP + FP)

4.1.2 Recall

The main objective of recall is to search for positive cases.
Recall = TP / (TP+FN)

4.1.3 F1 score

F1 score is the mean of precision and recall
F1 Score = 2*(Recall * Precision) / (Recall + Precision)

4.1.4 Support

Fundamental characteristic of support is the class within the indicated dataset is numerous really event by support

4.2 Experiment Results

A point to point explore amid the usage of the model has been performed. Stacking and neural network techniques are used here. It was understood from the technique of neural networks that the system may additionally be over-fitted. Developing a neural network based system needs to be considered by a number of parameters. Each optimizer algorithm to determine which parameter would perform best for the model. For this experiment here the number of hidden layers and the epoch's size are considered as HL and EPS that means HL5 indicates the number of hidden layers is 5 and the number of hidden layer 2 indicates HL2. Epochs size 50 that indicates EPS50. (S. S. M. M et al., 2021) According to the determination rules for this data set HL2 performs well but in this system HL5 performs better than HL2. Before final model formation the system has adjust with optimized parameters. Evaluating the optimized parameters accuracy, MSE, MAE are considered.

4.2.1 Here describe 5 adaptive algorithm of DNN

4.2.1.1 Adam Optimizer

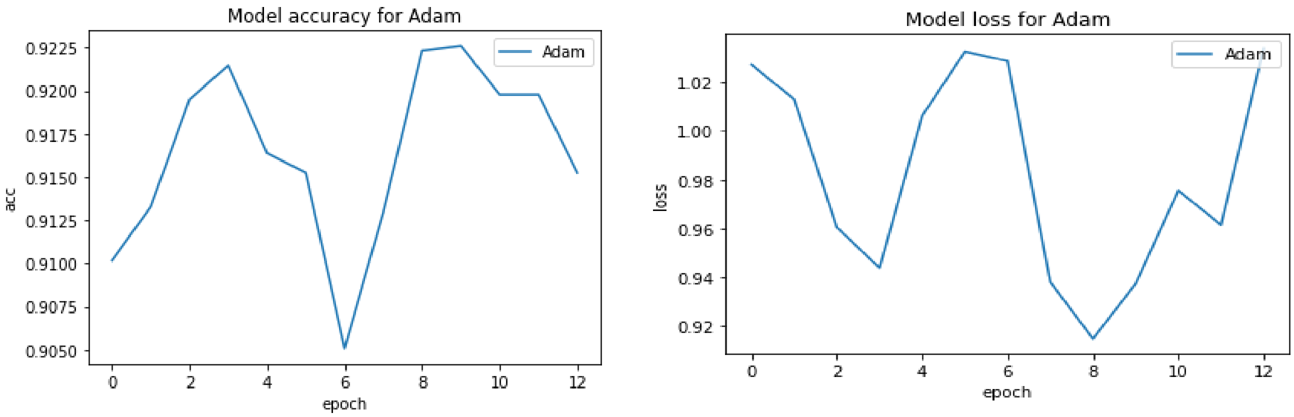

Figure 4. 1: Accuracy and loss of Adam Optimizer

Confusion Matrix for Adam optimizer

	Classified Phishing	Classified Legitimate
Actual Phishing	TP = 1976	FN = 78
Actual legitimate	FP = 86	TN = 1508

***Table 4.3:* Confusion Matrix for Adam optimizer**

4.2.1.2 SGD Optimizer

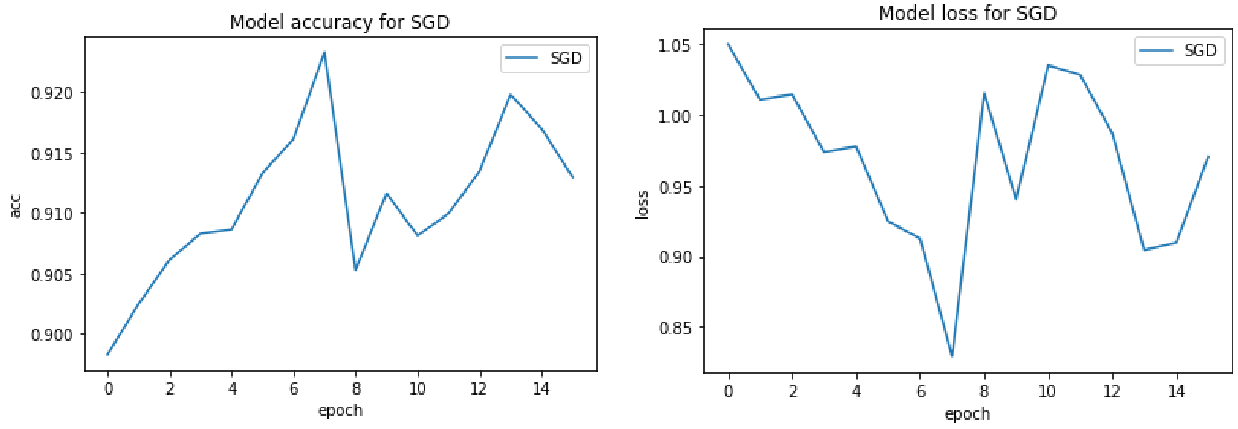

Figure 4.2: Accuracy and loss for SGD Optimizer

Confusion Matrix for SGD optimizer

	Classified Phishing	Classified Legitimate
Actual Phishing	TP = 1983	FN = 71
Actual legitimate	FP = 85	TN = 1509

Table 4.4: Confusion Matrix for SGD optimizer

4.2.1.3 RMSprop Optimizer

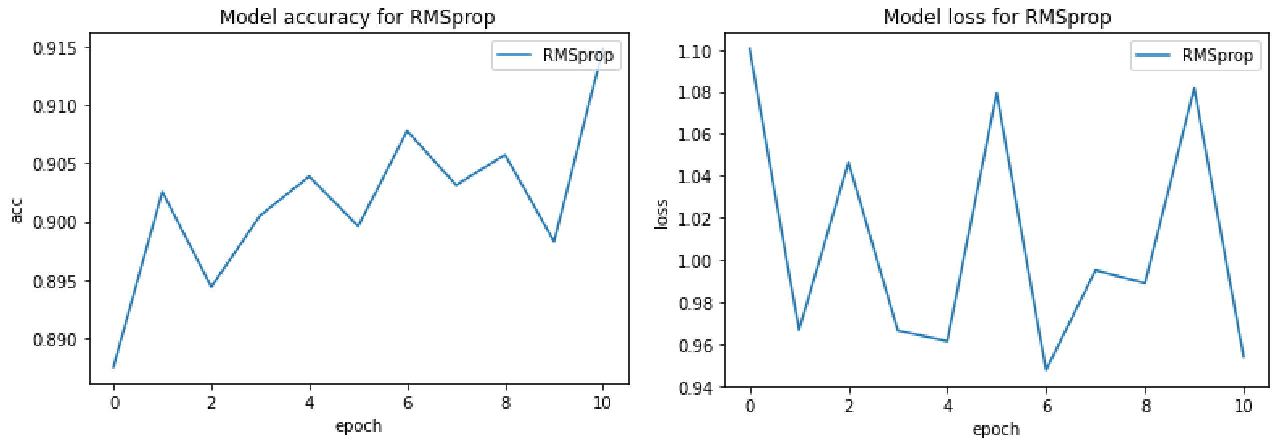

Figure 4.3: Accuracy and loss for RMSprop Optimizer

Confusion Matrix for RMSprop optimizer

	Classified Phishing	Classified Legitimate
Actual Phishing	TP = 1951	FN = 103
Actual legitimate	FP = 66	TN = 1528

Table 4.5: Confusion Matrix for RMSprop Optimizer

4.2.1.4 Adadelta Optimizer

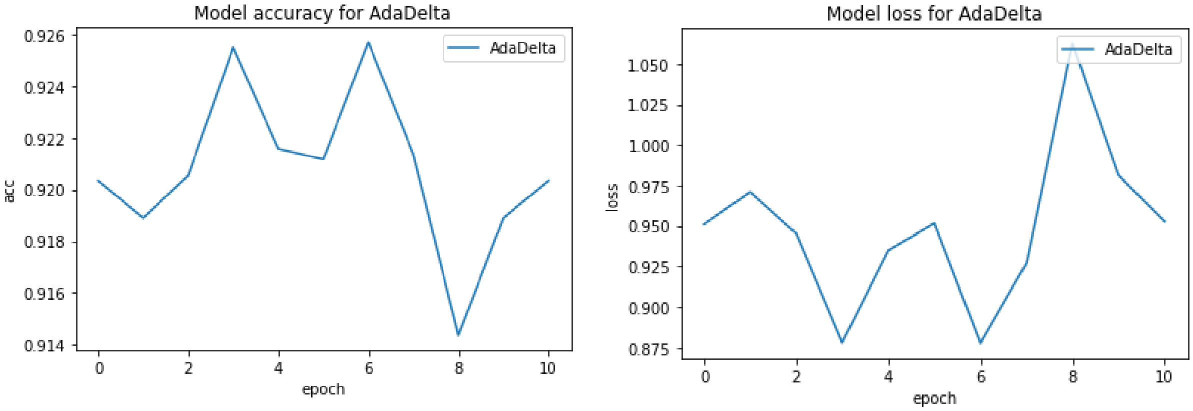

Figure 4.4: Accuracy and loss for AdaDelta Optimizer

Confusion Matrix for AdaDelta optimizer

	Classified Phishing	Classified Legitimate
Actual Phishing	TP = 1967	FN = 87
Actual legitimate	FP = 80	TN = 1514

Table 4. 6: Confusion Matrix for AdaDelta optimizer

4.2.1.5 Adagard Optimizer

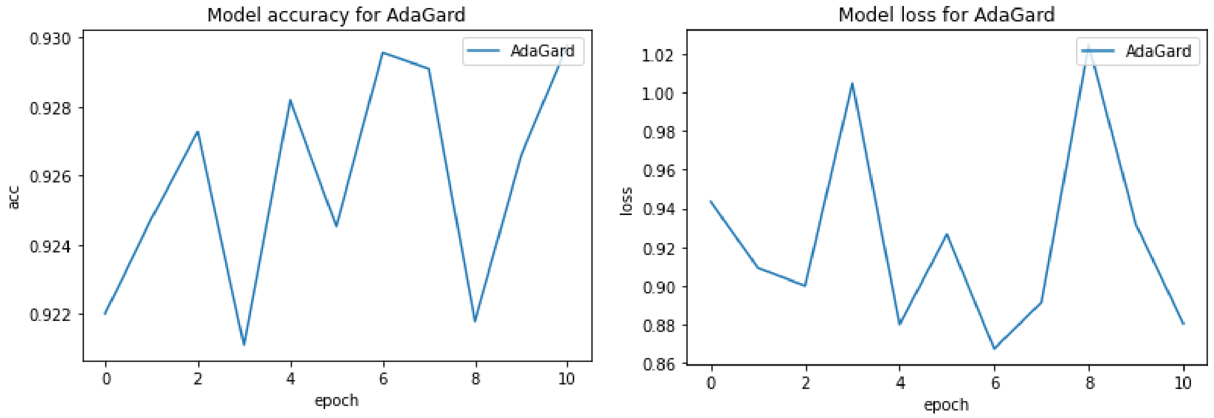

Figure 4. 5: Accuracy and Loss for AdaGard Optimizer

Confusion Matrix for AdaGard optimizer

	Classified Phishing	Classified Legitimate
Actual Phishing	TP = 1960	FN = 94
Actual legitimate	FP = 64	TN = 1530

Table 4. 7 : Confusion Matrix for AdaGard optimizer

Comparison among five optimizer with accuracy and loss

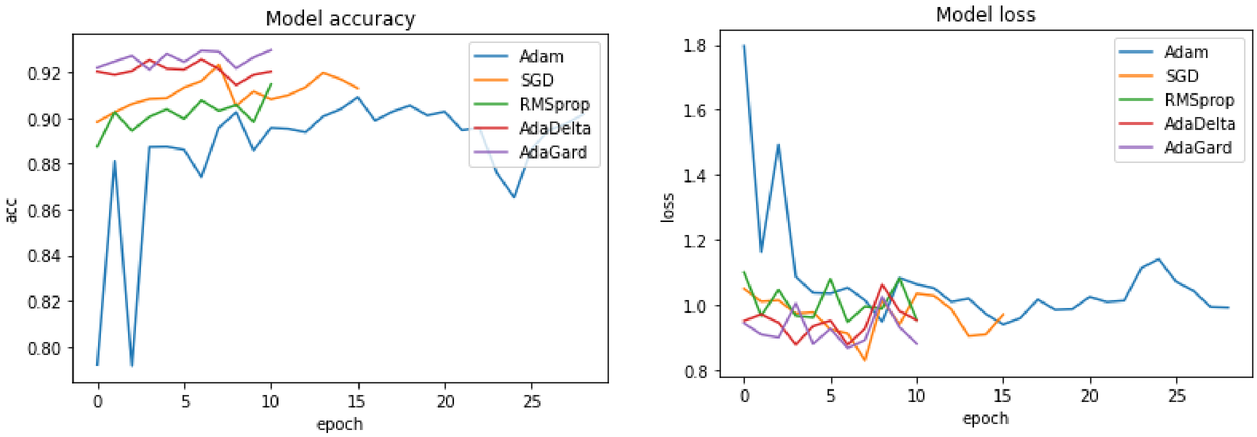

Figure 4.6: Accuracy and Loss for DNN Five Optimizer

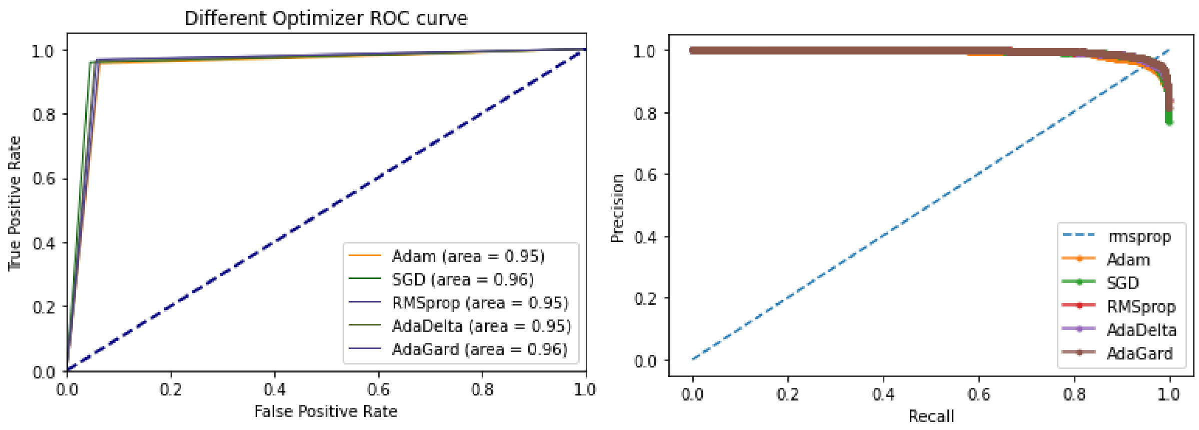

Figure 4.7: Different Optimizer for Roc Curve and Precision-Recall Curve

The ROC curve and precision-Recall curve the have been shown in figure 4.25 and 4.26. Maximum accuracy 0.955 attained from Adam individually. In case of precision-recall curve and the AUC-ROC curve SGD and AdaGard do better provides 0.96.SGD and AdaGard perform better in ROC curve and precision-Recall curve than others.

Evaluation comparison table for DNN

Serial	Optimizer	Label	Learning rate	Epochs	Accuracy	Mean squared error (MSE)	Mean absolute error (MAE)
1	Adam	HL5	0.01	50	0.955	0.030	0.074
2	SGD	HL5	0.001	100	0.951	0.021	0.049
3	RMSprop	HL5	0.0003	150	0.953	0.028	0.076
4	AdaDelta	HL5	0.0027570	250	0.954	0.023	0.078
5	AdaGard	HL5	0.0017470	150	0.953	0.018	0.049

Table 4. 8 : Evaluation comparison table for DNN

Accuracy, Val accuracy, loss, Val loss of the DNN five optimizers are shown through the line graph above. So that at a glance one can get an idea about all the optimizer algorithm reports. There is an individual line graph for all the optimizer algorithm and there is a line graph in the combination of all optimizers where all the optimizer are together, where is the comparison of all the optimizer algorithms result from there. According to the table 4.8 here provides the of DNN optimizers algorithm result. The above table shows the effect of the learning rate and epoch's size on the results of each optimizer algorithm. The better combination of various parameters increase the performance of the algorithm. According to Adam optimizer a performance evaluation was run

20 times for Adam optimizer. Different learning rates are used in that evaluation of 20 times, various results comes in these 20 times. Here is the best output from the 20 times run output that shows how much learning rate and how many epoch size are used for Adam optimizer. So in this case it can be said that in case of 5 layer DNN model using learning rate 0.01 and epoch size 50 provides maximum accuracy 0.955, MSE 0.030 and MAE 0.074. Similarly the performance of SGD optimizer algorithm is evaluated 20 times. Various learning rates are used in that evaluation of 20 times, various outcome comes in these 20 times. Here is the best output from the 20 times run output that shows how much learning rate and how many epoch size are used for SGD optimizer. So in this case it can be said that in case of 5 layer DNN model for SGD using learning rate 0.001 and epoch size 100 provides maximum accuracy 0.951, MSE 0.021 and MAE 0.049. Performance evaluation is run 20 times for RMSprop optimizer algorithm where 10 different learning rate are applied for 20 times and 20 different epoch size are use. So in this way it can be mentioned that in case of 5 layer DNN model (HL5) for RMSprop using learning rate 0.0003 and EPS150 (epoch size 150) provides maximum accuracy 0.953, MSE 0.028 and MAE 0.076. In the case of AdaDelta , 20 times the performance is evaluated to find a better combination of optimized parameters. Various learning rates and epochs size are used that any combination yields good outcome. Getting from the given 20 outcomes, learning rate 0.0027570, EPS250, accuracy 0.954, MSE 0.023 and MAE 0.078 using HL5, which is the best output combination for AdaDelta optimizer algorithm. According to AdaGard , 20 times the parameters are evaluated in various combinations for AdaGard for finding the better combination of parameters that provides the better outcome. After evaluating 20 time is then available for HL5 and EPS150 the desired accuracy is 0.953, MSE 0.018 and MAE is 0.049 learning rate0.0017470.

Observing all the outcomes from table 4.8 from above, it can be observe that all the optimizer provides 95 percent accuracy of which Adam pays a little more, Adam is the top scorer (Vrbančič et al., 2018). Random features provide training set imbalance results due to selection. Solving this problem some changes in coding have to be made so that the training set always train same features that there is no big difference in its results. Find the best combination of optimized parameters by turning on each optimizer algorithm performance evaluation 20 times and best of select from the given result for Adam optimizer HL5 and EPS50 the accuracy is 0.955, MSE 0.030 and MAE is 0.074.

From the above investigation have been conduct the confusion matrix of DNN about five deep learning adaptive optimizer namely **Adam, SGD, RMSporp, Adadelta, AdaGrad.** In this investigation **Table 4.3-4.7** among all optimizer **Adam** given the highest accuracy **95.5%** and **SGD** given the lowest accuracy **95.1%** for the deep neural network. On the Contrary, for the **MAE** lowest **0.49%** errors have been given for the **SGD** and **AdaGard** optimizer and **AdaDelta** given height **0.78%** errors in this investigation in Deep Neural Network. In this case researchers who want to work to detected phishing URL **Adam** optimizer will be the best optimizer for detect the Phishing URL more efficiently.

4.2.2 Adaptive Optimizer for Neural Network

4.2.2.1 Adam Optimizer

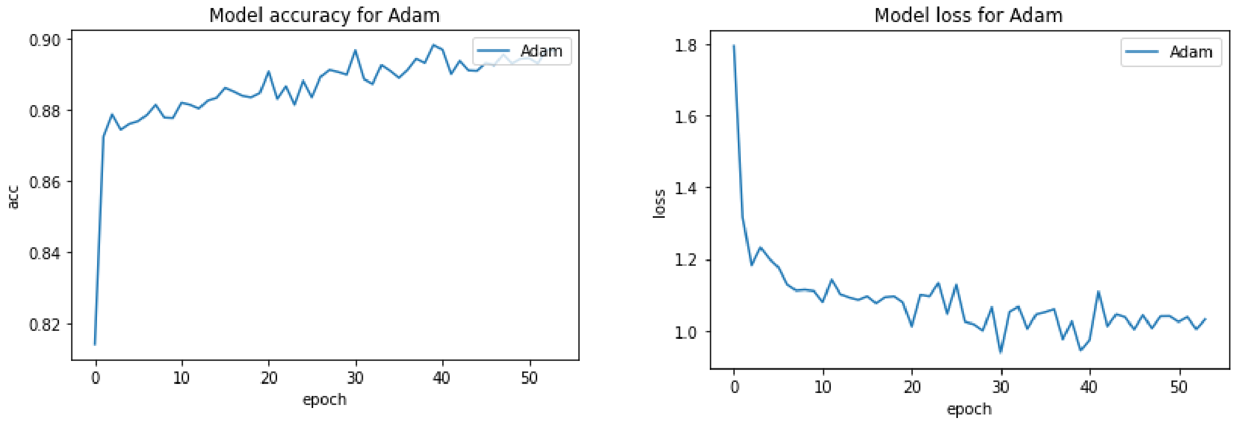

Figure 4. 8: Accuracy and Loss for Adam Optimizer

Confusion Matrix for Adam optimizer

	Classified Phishing	Classified Legitimate
Actual Phishing	TP = 1969	FN = 85
Actual legitimate	FP = 104	TN = 1490

Table 4. 9 : Confusion Matrix for Adam optimizer

4.2.2.2 SGD Optimizer

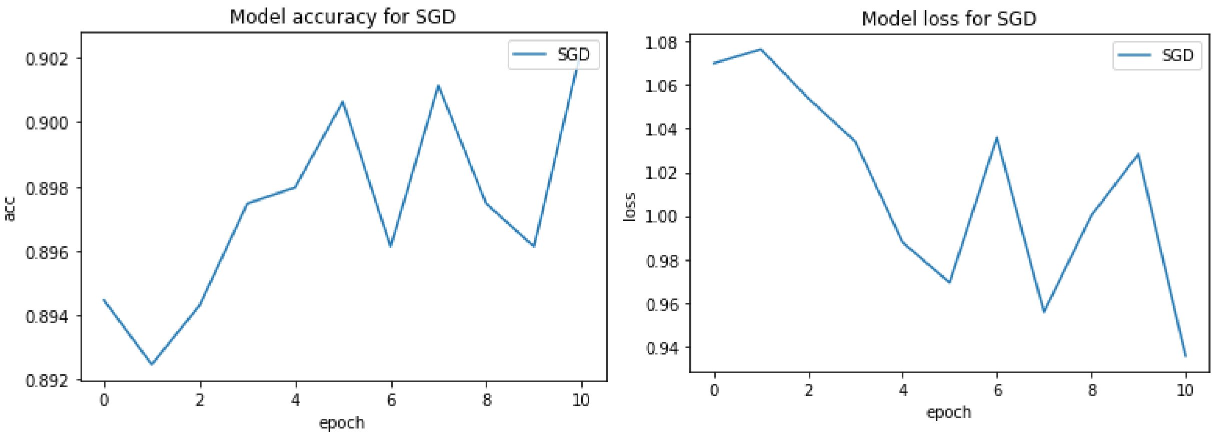

Figure 4.9: Accuracy and Loss for SGD Optimizer

Confusion Matrix for SGD optimizer

	Classified Phishing	Classified Legitimate
Actual Phishing	TP = 1968	FN = 86
Actual legitimate	FP = 108	TN = 1486

Table 4. 10 : Confusion Matrix for SGD optimizer

4.2.2.3 RMSprop Optimizer

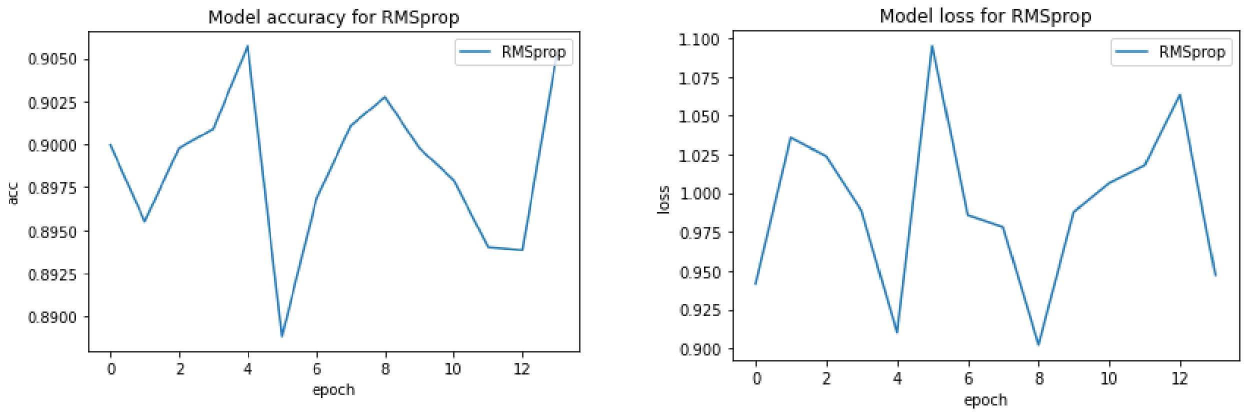

Figure 4.10: Accuracy and Loss For RMSprop Optimizer

Confusion Matrix for RMSprop optimizer

	Classified Phishing	Classified Legitimate
Actual Phishing	TP = 1957	FN = 97
Actual legitimate	FP = 98	TN = 1496

Table 4. 11 : Confusion Matrix for RMSprop optimizer

4.2.2.4 AdaDelta Optimizer

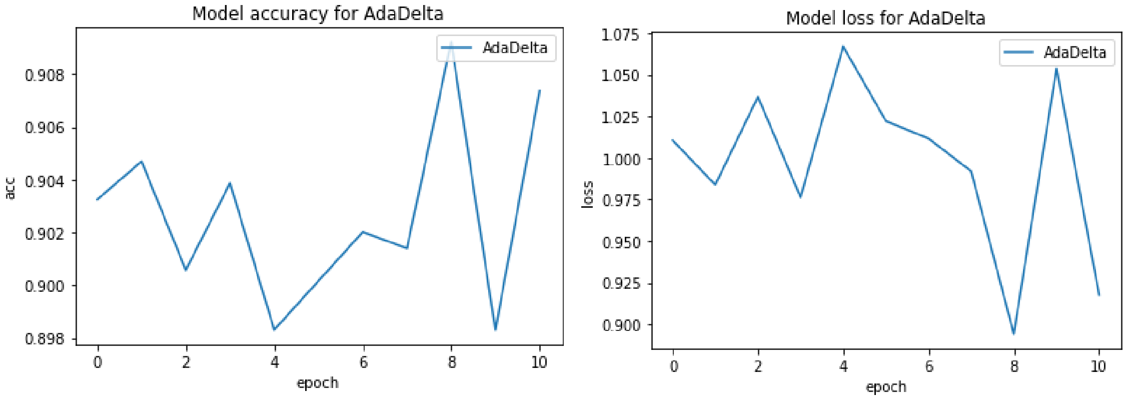

Figure 4.11: Accuracy and Loss for AdaDelta Optimizer

Confusion Matrix for AdaDelta optimizer

	Classified Phishing	Classified Legitimate
Actual Phishing	TP = 1964	FN = 90
Actual legitimate	FP = 103	TN = 1491

Table 4. 12 : Confusion Matrix for AdaDelta optimizer

4.2.2.5 AdaGard Optimizer

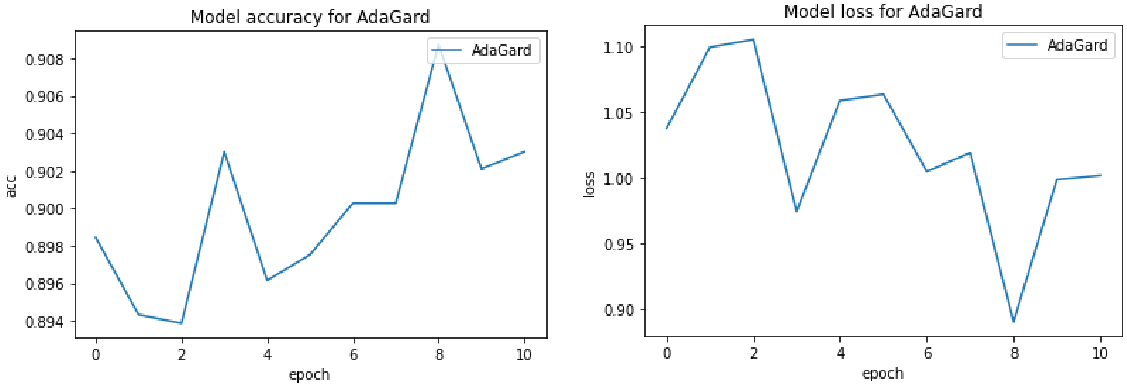

Figure 4.12: Accuracy and Loss for AdaGard Optimizer

Confusion Matrix for AdaGard optimizer

	Classified Phishing	Classified Legitimate
Actual Phishing	TP = 1977	FN = 77
Actual legitimate	FP = 105	TN = 1489

Table 4. 13 : Confusion Matrix for AdaGard optimizer

Comparison among five optimizer with accuracy and loss

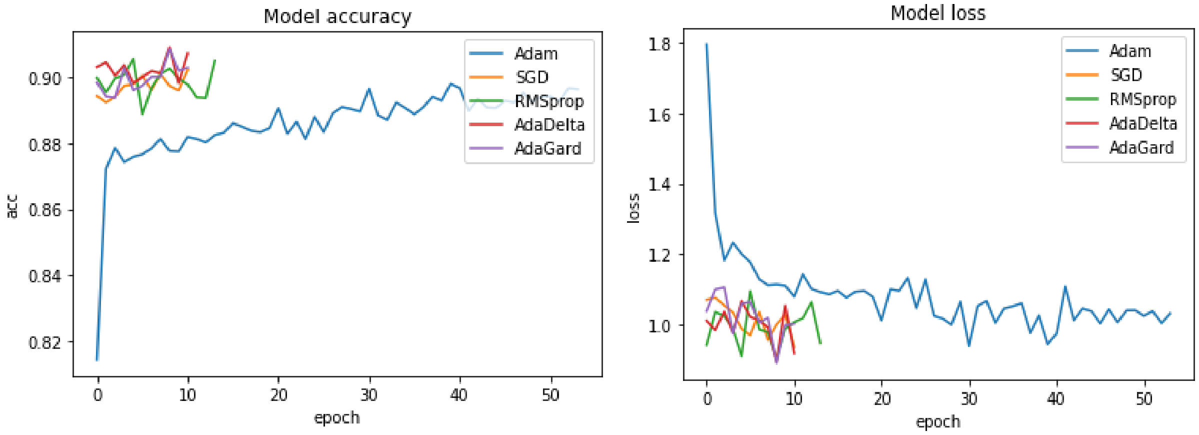

Figure 4.13 : Accuracy and Loss for DNN Five Optimizer

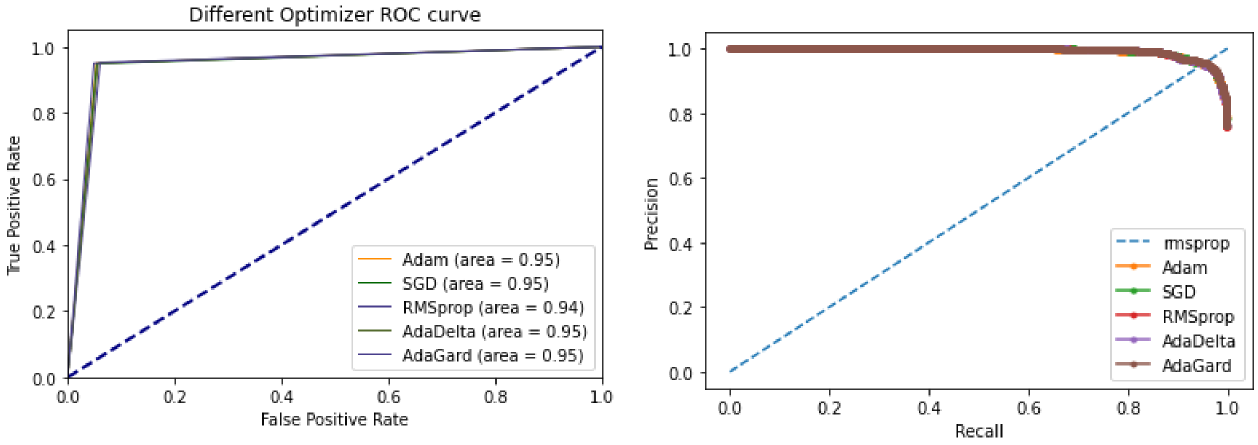

Figure 4.14: Different Optimizer Roc Curve and Precision-Recall Curve

The ROC curve and precision-Recall curve the have been shown in figure 4.51 and 4.52. Maximum accuracy 0.955 attained from AdaGard individually. In case of precision-recall curve and the AUC-ROC curve Adam, SGD, AdaDelta and AdaGard do better provides 0.95, expect RMSprop.

Evaluation comparison table for NN

Serial	Optimizer	Label	Learning rate	Epochs	Accuracy	Mean squared error (MSE)	Mean absolute error (MAE)
1	Adam	HL5	0.0017470	150	0.948	0.014	0.058
2	SGD	HL5	0.001	128	0.945	0.026	0.086
3	RMSprop	HL5	0.0003	200	0.948	0.026	0.080
4	AdaDelta	HL5	0.0027570	250	0.949	0.016	0.067
5	AdaGard	HL5	0.0017470	150	0.955	0.029	0.073

Table 4. 14: Evaluation comparison table for NN

An idea of how the adaptive optimizer algorithm work out using NN can be obtained by observed the above line graph. The main purpose of placing these line graph in the result section is to understand the outcome at a glance. Accuracy, Val accuracy, loss, Val loss of the NN five optimizers are shown through the line graph above. There is an individual line graph for all the optimizer algorithm and there is a line graph in the combination of all optimizers where all the optimizer are together, where is the comparison of all the optimizer algorithms result from there. According to the table 4.14 here provides the of NN optimizers algorithm result. The above table shows the effect of the learning rate and epochs size on the results of each optimizer algorithm. The better combination of various parameters increase the performance of the algorithm. According to Adam optimizer a performance evaluation was run 20 times for Adam optimizer. Different learning rates are used in that evaluation of 20 times, various results comes in these 20 times. Here is the best output from the 20 times run output that shows how much learning rate and how many epoch size are used for Adam optimizer. So in this case it can be represent that in case of HL5 NN model using learning rate 0.0017470 and epoch size 150 provides maximum accuracy 0.948, MSE 0.014 and MAE 0.058. Similarly the performance of SGD optimizer algorithm is evaluated 20 times. Various learning rates are used in that evaluation of 20 times, various outcome comes in these 20 times. Here is the best output from the 20 times run output that shows how much learning rate and how many epoch size are used for SGD optimizer. So in this case it can be represent that in case of HL5 and EPS128 for NN model for SGD using learning rate 0.001 provides maximum accuracy 0.945, MSE 0.026 and MAE 0.086. Performance evaluation is run 20 times for RMSprop optimizer algorithm where 10 different learning rate are applied for 20 times and 20 different epoch size are use. So in this way it can be mentioned that in case of HL5

NN model EPS200 for RMSprop using learning rate 0.0003 and provides maximum accuracy 0.948, MSE 0.026 and MAE 0.080. In the case of AdaDelta , 20 times the performance is evaluated to find a better combination of optimized parameters. Various learning rates and epochs size are used that any combination yields good outcome. Getting from the given 20 outcomes, learning rate 0.0027570 EPS250, accuracy 0.949, MSE 0.016 and MAE 0.067, which is the best output combination for AdaDelta optimizer algorithm. According to AdaGard , 20 times the parameters are evaluated in various combinations for AdaGard for finding the better combination of parameters that provides the better outcome. After evaluating 20 time is then available for HL5 and EPS150 the desired accuracy is 0.955, MSE 0.029 and MAE is 0.07.

Observing all the outcomes from Table 4.14 from above, it can be observe that all the optimizer provides 94 percent accuracy except AdaGard of which AdaGard pays a little more, AdaGard is the top scorer for NN . Random features provide training set imbalance results due to selection. Solving this problem some changes in coding have to be made so that the training set always train same features that there is no big difference in its results. Find the best combination of optimized parameters by turning on each optimizer algorithm performance evaluation 20 times and best of select from the given result for AdaGard optimizer HL5 and EPS150 the accuracy is 0.955, MSE 0.029 and MAE is 0.07 by learning rate0.0017470.

From the above investigation have been conduct the confusion matrix of NN about five deep learning adaptive optimizer namely **Adam, SGD, RMSporp, Adadelta, AdaGrad.** In this investigation **Table 4.9-4.13** among all optimizer **AdaGrad** given the highest accuracy **95.5%** and **SGD** given the lowest accuracy **94.5%** for the deep neural network. On the Contrary, for the **MAE** lowest **0.58%** errors have been given for the **Adam** and **SGD** optimizer given height **0.86%** errors in this investigation in Deep Neural Network. In this case researchers who want to work to detected phishing URL **AdaGrad** optimizer will be the best optimizer in the sphere of **NN** to detect Phishing URL more efficiently.

4.2.3 Stacking

The main purpose of stacked generalization is used a higher grade model to combine low grade models to achieve higher predictive accuracy. Stacking combines multiple model and learns it up for classification task.

Logistic Regression	Linear Discriminant Analysis	K neighbor	Decision Tee Classifier	Gaussian Naïve Bayes	Support Vector Machine
0.927	0.921	0.936	0.955	0.593	0.944

Table 4. 15 : Accuracy of Machine learning classifier Algorithm

According to table 4.16, first of all here 6 machine learning algorithms are used on the data of the desired dataset then some accuracy is found on the basis of that algorithm. These algorithm are used to build a stack model.

After applying stacking technique (Build model stack)

Logistic Regression	Linear Discriminant Analysis	K neighbor	Decision Tee Classifier	Gaussian Naïve Bayes	Support Vector Machine
0.966	0.965	0.965	0.966	0.965	0.966

Table 4. 16: Build model stack and the increased accuracy of Machine learning Algorithm

In this step a stack model is generated by applying these algorithm. Notice this table 4.17 this algorithm have changed in their accuracy after generating a stack model. The stack stipulates that it combines multiple models and learns for classification task. So purpose of this step is to stack learn stack.

Misclassification Rate

Random Forest	Decision Tree Classifier	Multilayer perception classifier	Support vector Machine	Stochastic Gradient	Gaussian Naïve Bayes
0.0047	0.0063	0.0022	0.0072	0.0073	0.012

Table 4. 17 : Misclassification rate

Accuracy

Random Forest	Decision Tree Classifier	Multilayer perception classifier	Support vector Machine	Stochastic Gradient	Gaussian Naïve Bayes
0.96	0.95	0.96	0.944	0.91	0.59

Table 4. 18 : Accuracy of Temporary Prediction

The stack has already been learned, now it knows how to process a model. Table 4.19 and table 4.18 represent the final accuracy and misclassification rate for first step. This work is done by two steps. So this table's value indicates the first step's prediction or temporary prediction because after second step prediction will find final prediction. The next step is to build a model, according to the study a model has been created XGBClassifire and through that model fitted the previous trained data and predict the final results.

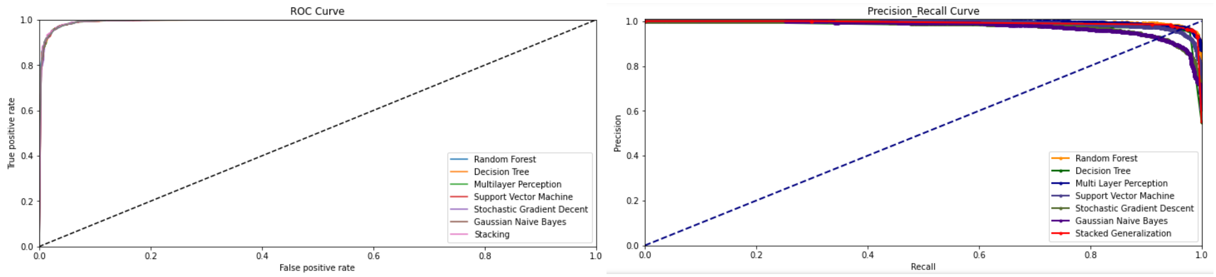

Figure 4.15: Different algorithms ROC Curve and Precision-Recall Curve

The precision-Recall curve and the ROC curve have been shown in figure 4.53 and 4.54.The first step shows that the maximum accuracy 0.96 with minimum error rate. RF and MLP do better individually where precision-recall curve and the AUC-ROC curve, stacked generalization performs low. However in the time of final prediction stack generalization provides accuracy 0.97.

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 Finding and Contributions

As phishing is a sensational phenomenon in today's online world, so it is a matter of concern in the current generation. Attackers carry out phishing attacks by sending various malicious URLs via email and social media platforms. Many techniques are available to detect phishing but some of these limitations are noted from previous study. In this study, anti-phishing techniques have been developed based on neural network, deep neural network and stacking technique. Parameter adjustment plays a vital role for these techniques, among these parameter learning rate is one of them. This is an unimaginable footstep of increasing the performance of neural network based on systems. Here is an assessment of the effect of parameters that will be an evidence in the creation of a neural network based system. The amount of data in the data set affects the system learning. In the case of stacking, Random Forest and Multilayer perception provides better results for precision and recall. However stack generalization helps better to enhance the overall accuracy.

This study dictates basically three multilayer techniques that are NN, DNN, stacking. Evaluating their performance shows that the results they provide are almost same among these DNN provides better accuracy. Here 2 layers are used for NN, 5 layers for DNN and stack generalization has used two steps. DNN and NN layers have units. These units indicate how deep these layers can go. Fundamental difference between NNN and DNN is that NN works with two layers on the behalf of DNN works with more than two layers. The value of unit basically indicates how depth the data will go and how many combinations will be tree based. A complete accurate outcome is obtained from multiple averages of a value. Stacking technique and NN provide good results for simple dataset, if the dataset holds complex or more complicated values then performance is likely to decrease. According to DNN, it works with a large number of layers and uses the value of the unit as needed. From this study, DNN model based architecture provides good results most of the time for any type of dataset.

5.2 Recommendation for Future Works

This study has worked on thirty features that can detect phishing websites from legal sites. But there are more features for phishing that are not mentioned here. In the imitation of the URLs pattern these thirty features are parted into four categories. Day by day more novel features are ejected. **In the future, adding more novel features and providing them by analyzing results. Which will provide a clear concept, from that the importance of each feature can be realized. As a result, future analysis on phishing detection will be much more understandable.**

REFERENCES

[1] Adebowale, M. A., Lwin, K. T., & Hossain, M. A. (2019, August). Deep Learning with Convolutional Neural Network and Long Short-Term Memory for Phishing Detection. In *2019 13th International Conference on Software, Knowledge, Information Management and Applications (SKIMA)* (pp. 1-8). IEEE.

[2] Acquisti, A., Adjerid, I., Balebako, R., Brandimarte, L., Cranor, L. F., Komanduri, S., ... & Wilson, S. (2017). Nudges for privacy and security: Understanding and assisting users' choices online. *ACM Computing Surveys (CSUR)*, *50*(3), 1-41.

[3] Adamson, A. S., & Smith, A. (2018). Machine learning and health care disparities in dermatology. *JAMA dermatology*, *154*(11), 1247-1248.

[4] Cui, Q., Jourdan, G. V., Bochmann, G. V., Couturier, R., & Onut, I. V. (2017, April). Tracking phishing attacks over time. In *Proceedings of the 26th International Conference on World Wide Web* (pp. 667-676)

[5] Dong, Z., Kapadia, A., Blythe, J., & Camp, L. J. (2015, May). Beyond the lock icon: real-time detection of phishing websites using public key certificates. In *2015 APWG Symposium on Electronic Crime Research (eCrime)* (pp. 1-12). IEEE.

[6] Das, R., Hossain, M., Islam, S., & Siddiki, A. (2019). *Learning a deep neural network for predicting phishing website* (Doctoral dissertation, Brac University).

[7] El-Alfy, E. S. M. (2017). Detection of phishing websites based on probabilistic neural networks and K-medoids clustering. *The Computer Journal*, *60*(12), 1745-1759.

[8] Fister, I., Suganthan, P. N., Kamal, S. M., Al-Marzouki, F. M., Perc, M., & Strnad, D. (2016). Artificial neural network regression as a local search heuristic for ensemble strategies in differential evolution. *Nonlinear Dynamics*, *84*(2), 895-914.

[9] Gupta, S., & Singhal, A. (2018). Dynamic classification mining techniques for predicting phishing URL. In *Soft Computing: Theories and Applications* (pp. 537-546). Springer, Singapore.

[10] Huang, Y., Yang, Q., Qin, J., & Wen, W. (2019, August). Phishing URL Detection via CNN and Attention-Based Hierarchical RNN. In *2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)* (pp. 112-119). IEEE.

[11] Kumar, M. S., & Indrani, B. (2020). Frequent rule reduction for phishing URL classification using fuzzy deep neural network model. *Iran Journal of Computer Science*, 1-9.

[12] Li, Y., Yang, Z., Chen, X., Yuan, H., & Liu, W. (2019). A stacking model using URL and HTML features for phishing webpage detection. *Future Generation Computer Systems*, *94*, 27-39.

[13] Le, H., Pham, Q., Sahoo, D., & Hoi, S. C. (2018). URLNet: Learning a URL representation with deep learning for malicious URL detection. *arXiv preprint arXiv:1802.03162*.

[14] Le, H., Pham, Q., Sahoo, D., & Hoi, S. C. (2018). URLNet: Learning a URL representation with deep learning for malicious URL detection. *arXiv preprint arXiv:1802.03162*.

[15] Maennel, O. M., & Matulevicius, R. A New Heuristic Based Phishing Detection Approach Utilizing Selenium Web-driver.

[16] Mohammad, R. M. A. (2016). *An Ensemble Self-Structuring Neural Network Approach to Solving Classification Problems with Virtual Concept Drift and its Application to Phishing Websites* (Doctoral dissertation, University of Huddersfield).

[17] Parthasarathy, G., Tomar, D. C., & Praisy, K. C. (2016). An Enhancement Of Association Classification Algorithm For Identifying Phishing Websites. *Indian Journal of Computer Science and Engineering*.

[18] Pompon, R., Walkowski, D., Boddy, S., & Levin, M. (2018). 2018 Phishing and Fraud Report: Attacks Peak During the Holidays. *F5 LABS*.

[19] Parekh, S., Parikh, D., Kotak, S., & Sankhe, S. (2018, April). A new method for detection of phishing websites: URL detection. In *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)* (pp. 949-952). IEEE.

[20] Rahman, S. S. M. M., Gope, L., Islam, T., & Alazab, M. (2021). IntAnti-Phish: An Intelligent Anti-Phishing Framework Using Backpropagation Neural Network. In *Machine Intelligence and Big Data Analytics for Cybersecurity Applications* (pp. 217-230). Springer, Cham.

[21] Rahman, S. S. M. M., Islam, T., & Jabiullah, M. I. (2020). PhishStack: Evaluation of Stacked Generalization in Phishing URLs Detection. *Procedia Computer Science, 167,* 2410-2418.

[22] Rao, R. S., & Ali, S. T. (2015, April). A computer vision technique to detect phishing attacks. In *2015 Fifth International Conference on Communication Systems and Network Technologies* (pp. 596-601). IEEE.

[23] Somesha, M., Pais, A. R., Rao, R. S., & Rathour, V. S. (2020). Efficient deep learning techniques for the detection of phishing websites. *Sādhanā*, *45*(1), 1-18.

[24] Sahingoz, O. K., Baykal, S. I., & Bulut, D. (2018). Phishing detection from urls by using neural networks. *Computer Science and Information Technology, 8*(17), 41-54.

[25] Vrbančič, G., Fister Jr, I., & Podgorelec, V. (2019). Parameter setting for deep neural networks using swarm intelligence on phishing websites classification. *International Journal on Artificial Intelligence Tools, 28*(06), 1960008.

[26] Vrbančič, G., Fister Jr, I., & Podgorelec, V. (2018, June). Swarm intelligence approaches for parameter setting of deep learning neural network: Case study on phishing websites classification. In *Proceedings of the 8th International Conference on Web Intelligence, Mining and Semantics* (pp. 1-8).

[27] Viktorov, O. (2017). *Detecting phishing emails using machine learning techniques* (Doctoral dissertation, Middle East University).

[28] Vazhayil, A., Vinayakumar, R., & Soman, K. P. (2018, July). Comparative study of the detection of malicious URLs using shallow and deep networks. In *2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT)* (pp. 1-6). IEEE.

[29] Winterrose, M. L., Carter, K. M., Wagner, N., & Streilein, W. W. (2020). Adaptive attacker strategy development against moving target cyber defenses. In *Advances in Cyber Security Analytics and Decision Systems* (pp. 1-14). Springer, Cham.

[30] Woogue, P. D. P., Pineda, G. A. A., & Maderazo, C. V. (2017). Automatic Web Page Categorization Using Machine Learning and Educational-Based Corpus. *International Journal of Computer Theory and Engineering*, *9*(6).

[31] Wang, Z. Q., & Wang, D. (2017, March). Recurrent deep stacking networks for supervised speech separation. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 71-75). IEEE.

[32] Yang, P., Zhao, G., & Zeng, P. (2019). Phishing website detection based on multidimensional features driven by deep learning. *IEEE Access*, *7*, 15196-15209.